

IOWA STATE UNIVERSITY

Digital Repository

Information Systems and Business Analytics
Publications

Information Systems and Business Analytics

12-2020

A mathematical programming approach for integrated multiple linear regression subset selection and validation

Seokhyun Chung
University Of Michigan

Young-Woong Park
Iowa State University, ywpark@iastate.edu

Taesu Cheong
Korea University

Follow this and additional works at: https://lib.dr.iastate.edu/isba_pubs



Part of the [Business Intelligence Commons](#), [Dynamical Systems Commons](#), [Harmonic Analysis and Representation Commons](#), and the [Management Information Systems Commons](#)

Recommended Citation

Chung, Seokhyun; Park, Young-Woong; and Cheong, Taesu, "A mathematical programming approach for integrated multiple linear regression subset selection and validation" (2020). *Information Systems and Business Analytics Publications*. 2.
https://lib.dr.iastate.edu/isba_pubs/2

This Article is brought to you for free and open access by the Information Systems and Business Analytics at Iowa State University Digital Repository. It has been accepted for inclusion in Information Systems and Business Analytics Publications by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

A Mathematical Programming Approach for Integrated Multiple Linear Regression Subset Selection and Validation

Seokhyun Chung^{*1}, Young Woong Park^{†2}, and Taesu Cheong^{‡3}

¹Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI, USA

²Ivy College of Business, Iowa State University, Ames, IA, USA

³School of Industrial Management Engineering, Korea University, Seoul, South Korea

July 28, 2020

Abstract

Subset selection for multiple linear regression aims to construct a regression model that minimizes errors by selecting a small number of explanatory variables. Once a model is built, various statistical tests and diagnostics are conducted to validate the model and to determine whether the regression assumptions are met. Most traditional approaches require human decisions at this step. For example, the user adding or removing a variable until a satisfactory model is obtained. However, this trial-and-error strategy cannot guarantee that a subset that minimizes the errors while satisfying all regression assumptions will be found. In this paper, we propose a fully automated model building procedure for multiple linear regression subset selection that integrates model building and validation based on mathematical programming. The proposed model minimizes mean squared errors while ensuring that the majority of the important regression assumptions are met. We also propose an efficient constraint to approximate the constraint for the coefficient t -test. When no subset satisfies all of the considered regression assumptions, our model provides an alternative subset that satisfies most of these assumptions. Computational results show that our model yields better solutions (i.e., satisfying more regression assumptions) compared to the state-of-the-art benchmark models while maintaining similar explanatory power.

Keywords: *Regression diagnostics, Subset selection, Mathematical programming*

1 Introduction

Regression analysis is one of the most popular forms of statistical modeling for analyzing the relationship between variables. Due to its interpretability and simplicity, *multiple (multivariate) linear regression* has been the most commonly used model for several decades, with a variety of fields employing it to handle prediction tasks. Multiple linear regression seeks to identify the most suitable linear relationship for several explanatory variables and a response variable. A multiple linear regression model for m explanatory variables can be represented by the linear equation

$$\underset{n \times 1}{\mathbf{b}} = \underset{n \times m}{\mathbf{A}} \underset{m \times 1}{\mathbf{x}} + \mathbf{e},$$

where \mathbf{A} represents the data matrix corresponding to m explanatory variables with n observations, \mathbf{b} denotes the data matrix corresponding to the values of the response variable, \mathbf{x} is the matrix of estimated parameters corresponding to the coefficients of the variables, and \mathbf{e} refers to the errors between the predictions of the linear model and the response variable. We further assume that data matrices \mathbf{A} and \mathbf{b} are already standardized with their respective means and standard deviations, and thus, without loss

^{*}seokhc@umich.edu

[†]ywpark@iastate.edu; corresponding author

[‡]tcheong@korea.ac.kr

of generality, we can remove the intercept from the model above. It then makes sense that \mathbf{x} should be chosen to minimize the error.

Subset selection is a procedure in which a subset of m explanatory variables is selected to reduce model complexity while maintaining explanatory power. Guyon and Elisseeff [14] and James et al. [16] demonstrate that a reduced subset can provide faster, more cost-effective predictors, along with a better understanding of the underlying process that generated the data. A reduced subset can also prevent over-fitting in the presence of too many irrelevant or redundant features [38] and can help users to more readily understand the results [18].

Many studies have been conducted on subset selection in the literature. Stepwise selection, including forward selection, backward elimination, and their combination, are the most popular algorithms thanks to their simple implementation and fast computing time. However, the solutions found by these algorithms are often low quality due to their greedy characteristics. Therefore, in order to improve subset selection, more complicated algorithms have been proposed. Meta-heuristic algorithms are presented in Zhang and Sun [39] (Tabu Search), Siedlecki and Sklansky [33] (genetic algorithm), and Hafiz et al. [15] (particle swarm). Subset selection procedures based on statistical or machine learning methods have also been carried out. For example, a Bayesian approach to subset selection is taken by Mitchell and Beauchamp [24], while George and McCulloch [11] suggest a selection procedure based on the superiority of each subset estimated from posterior probabilities given by Gibbs sampling. Genuer et al. [10] point out that random forests can be employed in a strategy that involves the ranking of explanatory variables, which provides insight for the selection of the variable(s). In addition, other research on machine learning algorithms for subset selection has been conducted by Castellano and Fanelli [8] (neural network), Rakotomamonjy [29] (support vector machine), and Zheng and Wang [40] (information entropy-based objective function). By assigning an L_1 penalty to coefficients, the least absolute shrinkage and selection operator (LASSO) is introduced by Tibshirani [35]. Although there have been a few decades since LASSO was first devised, it is still one of the most commonly used regression subset selection methods. Note that most of the methods above are heuristic, which means they do not necessarily give the subset with the minimum mean squared error (MSE).

Once a multiple linear regression model with a selected subset is established, the built model is validated through various statistical tests [27]. This validation is essential when the model is used as an explanatory model, which is the usual case for linear regression. For example, if a coefficient in multiple linear regression fails to pass the statistical significance test, the interpretation of data using the constructed model is not statistically justified. If the residual plots show violation of the regression assumptions, the validity of the model is hurt. It is thus crucial to building a model that achieves a small MSE as well as statistical validity.

1.1 Related work

Although the best subset selection problem can be formulated as mixed integer quadratic programming (MIQP) when ordinary least squares (OLS) is used, the statistical community has not paid it close attention because it requires significant computational time for practical implementation. However, mathematical programming approaches to find the best subset minimizing MSE have gained increased attention in the literature recently. Bertsimas and King [1] have pointed out that the computing power of mixed integer programming (MIP) solvers has increased at a rapid rate in the last 25 years, while also emphasizing the significant progress in the development of exact algorithms to solve integer programs. These recent developments have made integer programming a key method for finding the best subset. Konno and Yamamoto [22] introduce an MIQP formulation to find the best subset with a fixed number of variables for multiple linear regression with the minimum sum of squared error (SSE). Konno and Takaya [21] propose a multistep method to generate a nearly optimal solution in a shorter time. Based on the formulation of Konno and Yamamoto [22], Bertsimas et al. [3] suggest obtaining tight big- M values in the indicator constraint to improve computing speed. They also introduce an algorithm to achieve an initial solution for a warm-start, effectively reducing the computing time. The choice of a best subset for logistic regression via MIP using piecewise linear approximation is also presented in Sato et al. [31]. Models and algorithms that do not require the number of selected variables to be fixed have also been proposed. Miyashiro and Takano [25] introduce mixed integer second-order cone programming formulations that do not restrict the cardinality of the subsets. Miyashiro and Takano [26] set Mallows' C_p as the goodness of fit for the model and formulate the subset selection as MIP. In their work, Mallows'

C_p includes the number of selected variables and hence the cardinality of the subset need not be fixed when Mallows' C_p is used as a measure of the goodness of fit. Park and Klabjan [28] consider an MIQP formulation for subset selection when the shape of the data is not only ordinary (i.e., $m < n$) but also for high dimensional (i.e., $m > n$) variable selection when the number of selected variables is not fixed. They also provide a proof for big- M being an upper bound on coefficients and an efficient heuristic algorithm for cases when $m < n$. Note that these studies only consider several goodness-of-fit measures to obtain an optimal subset. Gómez and Prokopyev [12] recently developed a mixed-integer fractional programming model for criteria such as the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and the Hannan-Quinn Information Criterion (HQIC). They further strengthen their formulations by exploiting the normal equations underlying the optimization problem in computing.

While numerous studies have focused on regression subset selection, few have considered regression assumptions and diagnostics in their solution approach. Diagnostics are indeed an essential component in building a good explanatory regression model to explain the linear relationship between the response and explanatory variables. While regression models can be used to make predictions and be evaluated based on their prediction performance, in this study, we focus on improving the interpretability and explanatory power of a regression model through regression diagnostics. Bertsimas and King [1] suggest a bootstrap-based algorithmic approach that iteratively solves MIP problems to obtain a desirable model. They use penalized objective functions and constraints for sparsity, multicollinearity reduction, robustness to outliers, and statistical significance. Tamura et al. [34] propose MIQP for best subset selection while eliminating multicollinearity from linear regression models. Carrizosaa et al. [7] study a mathematical model for subset selection in order to construct a linear regression model with the significant coefficients and small multicollinearity by constraints performing shrinkage of the coefficients. Kim and Kim [19] consider group-wise multicollinearity in best subset selection framework and developed a modified discrete first-order algorithm to reduce multicollinearities in the selected subset. Recently, Bertsimas and Li [2] proposed an MIP-based framework based on Bertsimas and King [1] and accounted for significance and multicollinearity. They also propose an asymptotic normality result, which works independent of the regression normality assumption.

Our model differs from Bertsimas and King [1] and Carrizosaa et al. [7] in that our algorithm finds a solution that satisfies the tests and diagnostics with only one call to an MIP solver, whereas they use an iterative method that calls an MIP solver multiple times or heuristic observations for modeling. When the regression normality assumption is met, our approach for significance test is equivalent to the model of Bertsimas and Li [2]; our work has been conducted independently and simultaneously with Bertsimas and Li [2].

In this study, we propose a fully automated model building and validation procedure for multiple linear regression via a mathematical programming-based algorithm that minimizes MSE while guaranteeing both (i) the statistical significance of the regression coefficients and (ii) regression assumptions and diagnostics. The proposed model can replace the traditional iterative validation and diagnostics steps, and, with the help of lazy constraint technique, successfully returns the best subset with significant coefficients and a regression assumption-satisfying model in most cases. In addition, for the statistical significance test that is difficult to formulate as an easier linear constraint, we propose an explicit constraint to approximate the exact constraint for the coefficient t -test. The explicit constraint is formulated as a linear constraint and directly added to the model upfront to further reduce the solution time and size of the branch-and-bound tree explored. Our model is also capable of providing an alternative solution when there is no subset that satisfies all of the considered regression assumptions and tests.

1.2 Contribution

The contributions of our paper can be summarized as follows:

- We provide a mathematical programming-based algorithm that allows a regression model to be built that satisfies the majority of statistical tests and diagnostics. Note that it is not trivial to incorporate this model validation step into other popular subset selection approaches such as LASSO and stepwise selection. To the best of our knowledge, residual-based diagnostics are incorporated into the model for the first time in the literature. The experimental results show that our model identifies a subset that satisfies all of the considered tests and assumptions within a reasonable time while minimizing adjusted R squared.

While an MIP solver is used to solve the problem, our algorithm is unique in the sense that it does not require multiple iterative calls to the solver. The experimental results demonstrate that our algorithm significantly speeds up the solution search in comparison to existing iterative methods, resulting in the success of finding better solutions.

- We propose an efficient explicit constraint that significantly reduces the solution space. The constraint is a relaxed version of the constraint we propose for the regression coefficient t -test. This relaxed constraint is added to the model before the branch-and-bound algorithm to exclude the solutions (subsets) that significantly violate the t -test from the branch-and-bound search tree. The computational experiment shows that the relaxed constraint reduces the solution time and size of the search tree.
- We present a logical procedure to find a near-feasible solution when no subset satisfies the tests and diagnostics or our model fails to find a feasible solution within the given time limit. To the best of our knowledge, our work is the first attempt that provides good alternative solutions in the case, which indeed often arises in practice. The proposed procedure mimics the typical steps used to build a linear regression model. The experimental results show that our procedure produces higher quality subsets than do the benchmarks.

1.3 Structure of the paper

The paper is structured as follows. In Section 2, we discuss several characteristics of a desirable multivariate linear regression model that incorporates transformed variables and meets essential regression assumptions. In Section 3, a mathematical programming approach for the best subset which reflects the features discussed in Section 2 is presented. A logical procedure for obtaining an alternative solution when the model is infeasible is proposed in Section 4. The results of computational experiments are presented in Section 5, followed by conclusions in Section 6.

2 Model Validation for Multiple Linear Regression

Subset selection in multiple linear regression aims to find a subset of explanatory variables that gives small fitting errors. At the same time, the model needs to satisfy the assumptions of linear regression from a statistical perspective. To find such subset, Neter et al. [27] suggests building a regression model by repetitively checking the assumptions and diagnostics. Figure 1 summarizes this strategy. After preprocessing the collected data, several explanatory variables are selected to build a statistically significant model. Diagnostics are then conducted on the model. If the model satisfies all assumptions, it is forwarded to the postprocessing step. Otherwise, another subset of explanatory variables is selected and tested. As a result, constructing a useful regression model requires many diagnostics tests. In this section, we discuss three popular techniques and tests that will be included in the mathematical formulation proposed in Section 3.

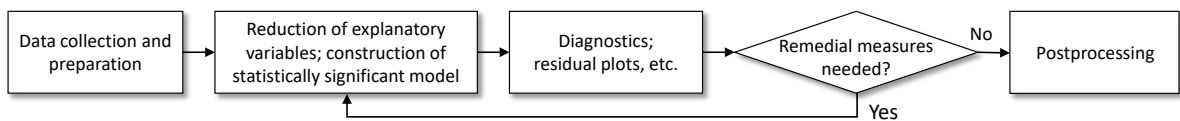


Figure 1: Strategy for linear regression model construction [27]

2.1 Transformation of explanatory variables

When the explanatory and response variables have a non-linear relationship, the non-linear trend can be observed in the residual plot. To remedy this issue, log or other nonlinear transformation can be used for the explanatory variables. When the transformed variables are added to the set of explanatory variables, at most one between the original and transformed explanatory variables can be included in the subset simultaneously to avoid multicollinearity. If the transformed explanatory variables work properly, the non-linear relationship issue in the residual plot can be fixed. Therefore, we generate transformed data

as a fixed candidate set of selected variables. The MIQP model presented in Section 3.2 has a constraint that selects at most one of the original and transformed variables from the pair.

2.2 Statistical tests for linear regression parameters

One of the popular tests in the model evaluation step is the statistical significance test for regression coefficients. This test checks if the estimated regression coefficients are non-zero using Student's t -distribution. Let $s(\mathbf{A})_j$ and \hat{x}_j be the standard deviation and estimated coefficients, respectively, of explanatory variable $j \in \{1, \dots, k\}$. Then, the coefficients follow Student's t -distribution.

$$\frac{\hat{x}_j - x_j}{s(\mathbf{A})_j} \sim t_{1-\frac{\alpha}{2}, n-k-1}, \quad j = 1, \dots, k. \quad (1)$$

Our interest is to test whether coefficient \hat{x}_j is equal to zero. We conclude that linear relationship between the response variable and the j -th explanatory variable is not statistically significant if \hat{x}_j is close to zero with a large standard deviation. On the other hand, if \hat{x}_j is large enough, we conclude that there is a significant linear relationship. Formally, the following hypothesis can be tested: $H_0 : x_j = 0, H_1 : x_j \neq 0$. Thus, to reject the null hypothesis, the following inequality must hold:

$$\left| \frac{\hat{x}_j}{s(\mathbf{A})_j} \right| \geq t_{1-\frac{\alpha}{2}, n-k-1}, \quad j = 1, \dots, k. \quad (2)$$

We will discuss how this requirement can be handled with a mathematical programming approach in Section 3.3 and Section 3.4.

2.3 Model validation with residual plots

One of the key assumptions of linear regression is that errors have constant variance over fitted values or over explanatory variables. This can be verified by drawing residual plots. The residuals have constant variance if they form a band-like region with constant width over the entire range of fitted values. However, linear models often do not satisfy the assumption. The violation of residual assumptions can induce the biased estimation of standard errors that may result in invalid inferences [6]. Thus, once a linear regression model is constructed, it is critical to check the diagnostic plots of the residuals.

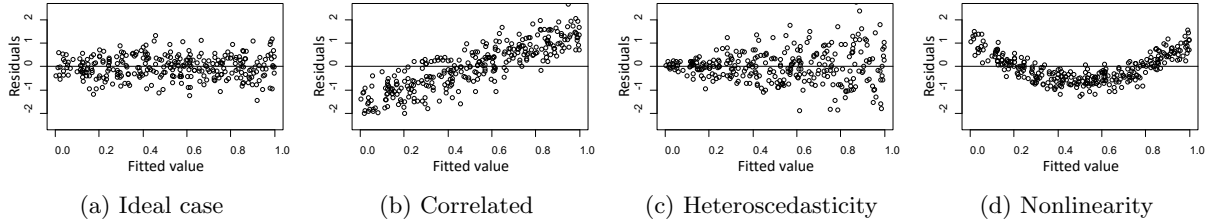


Figure 2: Examples of residual plot

Figure 2 shows representative plots of residuals versus fitted values. In Figure 2a, the variance of the residuals seems to be constant over the range of fitted values and thus the generated model satisfies the key regression assumption. However, this is not the case for the other examples in Figure 2. Figure 2b displays a positive *correlation* between the residuals and fitted values, Figure 2c shows *heteroscedasticity*, in which the variance of the residuals increases as the fitted values increase, and Figure 2d presents a *non-linear* relationship between the residuals and fitted values. The plot in Figure 2a is ideal, whereas the latter three cases in Figure 2 are problematic, need to be fixed, and frequently observed when implementing linear regression models for real-world problems. We next describe how to detect these cases without the visual aids.

Correlation in residuals To detect the situation depicted in Figure 2b, a simple linear regression model, called an *auxiliary model*, is constructed where the explanatory and response variables are the fitted values and residuals in the plot, respectively. If the estimated slope is close to zero, then we conclude the model does not suffer from linearity. If the estimated slope is far from zero, then we

conclude the model violates the residual assumption. A simple hypothesis test can be used to check whether the estimated slope is zero. To describe this test mathematically, let $\hat{\mathbf{b}}$ and $\hat{\mathbf{e}}$ be the vector of fitted values and residuals, respectively. Then, we establish an auxiliary model $\hat{\mathbf{e}} = x_1^{\text{aux}}\mathbf{1} + x_2^{\text{aux}}\hat{\mathbf{b}} + \boldsymbol{\epsilon}$ where x_1^{aux} and x_2^{aux} represent the estimated intercept and coefficient, respectively, and $\boldsymbol{\epsilon}$ is the error of the model. To check whether x_2^{aux} is statistically significant, the following hypothesis test is conducted: $H_0 : x_2^{\text{aux}} = 0$ and $H_1 : x_2^{\text{aux}} \neq 0$. It is then followed by a usual t -test. Note that a failure to reject H_0 indicates the linearity assumption is met.

Heteroscedasticity To detect heteroscedasticity (Figure 2c), two statistical tests need to be performed. The first test, referred to as an *absolute residual test* in this paper, is identical to the previous test except for the fact that the values of the response variable in the auxiliary model are the absolute values of the residuals. Observe that if the negative side of Figure 2c is flipped to overlap the positive side, an increasing trend in the residuals is observed as illustrated in Figure 3b. On the other hand, the same procedure for the ideal case of Figure 2a creates the constant variance of the absolute residuals over fitted values in Figure 3a. Thus, if a linear trend is detected from the auxiliary model of the absolute residuals, we can assume that the residuals of the linear model have heteroscedasticity. Another widely used diagnostic tool for heteroscedasticity is the *Breusch-Pagan test* [6]. To prevent being overly rigorous, we conclude that the heteroscedasticity exists and the residual assumption is violated only if both proposed tests indicate heteroscedasticity. In Section 3.4, our mathematical programming-based diagnostic approach is presented to capture the features of heteroscedasticity.

Nonlinearity in residuals To remedy the nonlinear trend in the residual plot in Figure 2d, we consider both original and transformed explanatory variables, as explained in Section 2.1, while we restrict the model to select at most one among the original and transformed variables.

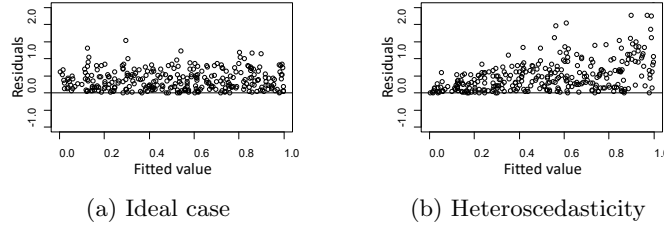


Figure 3: Examples of absolute residual plot

We note here that other tests such as the F -test for the model and the multicollinearity test for the pairwise coefficients can be incorporated into our framework proposed in Section 3.

3 Integrated Model Building and Validation Procedure via Mathematical Programming

In this section, we develop an automated procedure based on mathematical programming models that minimizes SSE to select a subset of a fixed number of explanatory variables and include statistical tests and diagnostics for multivariate linear regression. In Section 3.1, we introduce the base model from the literature, which does not consider statistical test or diagnostics. In Section 3.2, we extend the base model to include log-transformed explanatory variables. In Section 3.3, we propose constraints to remove statistically insignificant coefficients. In Section 3.4, we present the final algorithm, which considers all of the remaining tests and diagnostics.

3.1 Base model

We start with a basic model similar to that proposed by Wilson [36] and all data matrices are standardized. The sets, parameters, and decision variables used are as follows.

Sets and Parameters

(n, m) : number of observations and explanatory variables

$(\mathcal{I}, \mathcal{J})$: index set of observations, $\mathcal{I} = \{1, \dots, n\}$, and explanatory variables, $\mathcal{J} = \{1, \dots, m\}$
 k : number of selected explanatory variables
 \mathbf{A} : standardized data matrix corresponding to explanatory variables, $\mathbf{A} = [\mathbf{A}_j] = [a_{ij}] \in \mathbb{R}^{n \times m}$
 \mathbf{b} : standardized data matrix corresponding to response variables, $\mathbf{b} = [b_i] \in \mathbb{R}^n$

Decision Variables

x_j : coefficient of the j -th explanatory variable, $\forall j \in \mathcal{J}$
 e_i : error (i.e., residual) between the i -th observation and its prediction value, $\forall i \in \mathcal{I}$
 z_j : 1 if the j -th explanatory variable is selected; 0 otherwise, $\forall j \in \mathcal{J}$

Using the parameters and decision variables above, the following basic mathematical programming model can be formulated.

$$\text{minimize} \quad \sum_{i \in \mathcal{I}} e_i^2 \quad (3a)$$

$$\text{subject to} \quad e_i = \sum_{j \in \mathcal{J}} a_{ij} x_j - b_i, \quad \forall i \in \mathcal{I}, \quad (3b)$$

$$-Mz_j \leq x_j \leq Mz_j, \quad \forall j \in \mathcal{J}, \quad (3c)$$

$$\sum_{j \in \mathcal{J}} z_j = k, \quad (3d)$$

$$x_j \text{ unrestricted}, \quad z_j \in \{0, 1\}, \quad \forall j \in \mathcal{J}, \quad (3e)$$

$$e_i \text{ unrestricted}, \quad \forall i \in \mathcal{I}. \quad (3f)$$

The basic model (3) minimizes the SSE of a multivariate linear regression model with a fixed k . Note that although $\text{MSE} = \frac{\sum_{i \in \mathcal{I}} e_i^2}{n-k-1}$ is not directly minimized in (3a), minimizing $\text{SSE} = \sum_{i \in \mathcal{I}} e_i^2$ in (3a) is equivalent to minimizing MSE because k is fixed as a given parameter. Constraint (3b) defines the residuals, and constraint (3c) indicates that if an explanatory variable is not selected, then the coefficient of the variable must be zero. Lastly, constraint (3d) ensures that the number of selected variables is k .

The mathematical model (3) can be converted into an MIQP using a popular linearization technique. Instead of unrestricted continuous decision variables x_j and e_j , non-negative variables x_j^+ , x_j^- , e_j^+ , and e_j^- are used, where $e_i = e_i^+ - e_i^-$ and $x_i = x_i^+ - x_i^-$. By plugging these in, the following MIQP can be obtained:

$$\text{minimize} \quad \sum_{i \in \mathcal{I}} (e_i^{+2} + e_i^{-2})$$

$$\text{subject to} \quad e_i^+ - e_i^- = \sum_{j \in \mathcal{J}} a_{ij} (x_j^+ - x_j^-) - b_i, \quad \forall i \in \mathcal{I},$$

$$x_j^+ + x_j^- \leq Mz_j, \quad \forall j \in \mathcal{J}, \quad (4)$$

$$\sum_{j \in \mathcal{J}} z_j = k,$$

$$x_j^+, x_j^- \geq 0, \quad z_j \in \{0, 1\}, \quad \forall j \in \mathcal{J},$$

$$e_i^+, e_i^- \geq 0, \quad \forall i \in \mathcal{I}.$$

Recall that the main purpose of our study is to build a multivariate linear regression model that considers diagnostics. To meet this goal, we now extend the base mathematical model (4) to include the important diagnostic tests presented in Section 2 as constraints.

3.2 Inclusion of log-transformed explanatory variables

We first discuss how to include log-transformed explanatory variables in the model. Because an explanatory variable and its log-transformation are highly correlated, we need to prevent both variables from being selected simultaneously, as discussed in Section 2.1. Let us then define new parameters and sets.

Set and Parameters

\mathcal{J}^l : index set of the logarithm of the explanatory variables, $\mathcal{J}^l = \{m+1, \dots, 2m\}$

\mathbf{A}^l : standardized data matrix formed of the logarithm of the explanatory variables, $\mathbf{A}^l = [a_{ij}^l] \in \mathbb{R}^{n \times m}$

$\tilde{\mathbf{A}}$: data matrix concatenating \mathbf{A} and \mathbf{A}^l , i.e., $\tilde{\mathbf{A}} = [\mathbf{A} \ \mathbf{A}^l] = [\tilde{a}_{ij}] \in \mathbb{R}^{n \times 2m}$

If a vector corresponding to the j -th column of \mathbf{A} possesses non-positive elements, we compute the logarithm of the column using a conventional method to deal with the non-positives: $a_{ij}^l = \log(a_{ij} + |\min(\{a_{ij} | i \in \mathcal{I}\})| + 1)$. Because all original explanatory variables have transformed variables, we can define $j^l = j + m$ for each $j^l \in \mathcal{J}^l$. The model incorporating the log-transformed explanatory variables, $\mathcal{MP}_{\text{base}}(k)$, can be formulated as follows:

$$\mathcal{MP}_{\text{base}}(k) : \quad \text{minimize} \quad \sum_{i \in \mathcal{I}} (e_i^{+2} + e_i^{-2}) \quad (5a)$$

$$\text{subject to} \quad e_i^+ - e_i^- = \sum_{j \in \mathcal{J} \cup \mathcal{J}^l} \tilde{a}_{ij} (x_j^+ - x_j^-) - b_i \quad \forall i \in \mathcal{I}, \quad (5b)$$

$$x_j^+ + x_j^- \leq M z_j, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l, \quad (5c)$$

$$\sum_{j \in \mathcal{J} \cup \mathcal{J}^l} z_j = k, \quad (5d)$$

$$z_j + z_{j^l} \leq 1, \quad \forall j \in \mathcal{J}, j^l = j + m, \quad (5e)$$

$$x_j^+, x_j^- \geq 0, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l, \quad (5f)$$

$$e_i^+, e_i^- \geq 0, \quad \forall i \in \mathcal{I}, \quad (5g)$$

$$z_j \in \{0, 1\}, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l. \quad (5h)$$

The extended model $\mathcal{MP}_{\text{base}}(k)$ is similar to formulation (4). The difference is that (5) has additional derived explanatory variables in \mathcal{J}^l and (5e) ensures that both the original and log transformed explanatory variables are not selected at the same time. We note that, in addition to the log-transformation, other types of transformation of the explanatory variables can be considered in a similar manner. Finally, we remark that an appropriate value of M in (5c) is required to solve the problem. When M is too small, we cannot guarantee optimality. When M is too large, the optimization solver can struggle due to numerical issues. Park and Klabjan [28] proposed a sampling-based approach, where M is estimated by iteratively sampling a subset of explanatory variables. We use this method with a slight modification to ensure that the sampled explanatory variables do not simultaneously include an original explanatory variable and its transformation.

3.3 Inclusion of constraints corresponding to t -tests for the significance of the regression coefficients

In this section, we present constraints to check the statistical significance of the regression coefficients. Formulating these constraints is not a trivial task because the statistical significance of an estimated coefficient depends on the selected subset. In detail, $s(\mathbf{A})_j$ in (1) can only be calculated given a subset so that an inequality including $s(\mathbf{A})_j$ cannot be trivially formulated as a convex constraint. In order to address this issue, we convert the inequality (2) into a constraint that checks the statistical significance of the j -th variable if it is selected. We derive

$$\left| \frac{x_j}{s(\mathbf{A}_{s(k)})_j} \right| \geq t_{1-\frac{\alpha}{2}, n-k-1} \quad (6a)$$

$$\implies |x_j| \geq t_{1-\frac{\alpha}{2}, n-k-1} |s(\mathbf{A}_{s(k)})_j| - M(1 - z_j) \quad (6b)$$

$$\implies x_j^+ + x_j^- \geq t_{1-\frac{\alpha}{2}, n-k-1} s(\mathbf{A}_{s(k)})_j - M(1 - z_j), \quad (6c)$$

where $s(k)$ is a subset with k explanatory variables, $\mathbf{A}_{s(k)}$ is a submatrix of \mathbf{A} derived from $s(k)$, and $s(\mathbf{A}_{s(k)})_j$ is the standard deviation of the estimated coefficient for explanatory variable j which is

equivalent to $\sqrt{\text{MSE}_{\mathbf{A}_{s(k)}}(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1}}$ and only defined for explanatory variables in $s(k)$ (i.e., $j \in s(k)$), but we omit the notation j from $s(k)$ for notational simplicity). The resulting constraint above is only activated when the j -th variable is selected (i.e., $z_j = 1$). Note that $\mathbf{A}_{s(k)}$ varies according to the selected variables. It implies that the values $s(\mathbf{A}_{s(k)})_j$ in the constraint above vary accordingly and we cannot calculate the $s(\mathbf{A}_{s(k)})_j$ value if subset $s(k)$ is not fixed. Hence, it is still difficult to add constraint (6c) in its current form to the MIQP model.

To handle changing values of $s(\mathbf{A}_{s(k)})_j$ depending on the selected subset, we use the lower bound of $s(\mathbf{A}_{s(k)})_j$, which gives a relaxed version of the exact constraint. A tight lower bound for $s(\mathbf{A}_{s(k)})_j$ can cut some of the subsets with insignificant regression coefficients. Also, the lower bound should be calculated very efficiently to avoid additional computational cost.

Let $R^2(\mathbf{A}, \mathbf{b})$ denote the coefficient of determination for a multiple linear regression model fitted to data matrices \mathbf{A} and \mathbf{b} . To obtain the lower bound for $s(\mathbf{A}_{s(k)})_j$, we start with the following definition.

Definition 1. In a multiple linear regression with design matrix \mathbf{A} , the variance inflation factor (VIF) of the j -th explanatory variable, denoted by VIF_j , is given as $VIF_j = \frac{1}{1-R^2(\mathbf{A}_{-j}, \mathbf{A}_j)}$ where \mathbf{A}_{-j} represents a submatrix of \mathbf{A} formed by excluding the j -th column.

Note that $R^2(\mathbf{A}_{-j}, \mathbf{A}_j)$ is the coefficient of determination for a linear regression model, where variable j is the response variable and all variables except for j are explanatory variables. The quantity VIF_j is indeed given by the j -th diagonal elements of $(\mathbf{A}'\mathbf{A})^{-1}$. Now we recall the lemma from Rencher and Schaalje [30].

Lemma 1. In a multiple linear regression model, if $s \subset s'$, then $R^2(\mathbf{A}_s, \mathbf{b}) \leq R^2(\mathbf{A}_{s'}, \mathbf{b})$.

Lemma 1 states that the coefficient of determination does not decrease by adding a new explanatory variable. Based on the lemma, we derive the lower bound for $s(\mathbf{A}_{s(k)})_j$ to obtain the relaxed constraints.

Theorem 1. Let $R^2_{-j} = \min_{j' \in (\mathcal{J} \cup \mathcal{J}^l) \setminus \{j\}} R^2(\mathbf{A}_{j'}, \mathbf{A}_j)$ for $j \in \mathcal{J} \cup \mathcal{J}^l$ and $k \geq 2$. Then for each j , $s(\mathbf{A}_{s(k)})_j \geq \sqrt{\frac{\text{MSE}_{\text{LB}}(k)}{1-R^2_{-j}}}$, where $\text{MSE}_{\text{LB}}(k)$ is the optimal solution of the LP-relaxation of $\mathcal{MP}_{\text{base}}(k)$.

Proof. Because $s(\mathbf{A}_{s(k)})_j = \sqrt{\text{MSE}_{\mathbf{A}_{s(k)}}(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1}}$, we can calculate the lower bound of $s(\mathbf{A}_{s(k)})_j$ by obtaining lower bounds of $\text{MSE}_{\mathbf{A}_{s(k)}}$ and $(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1}$.

We first derive a lower bound of $\text{MSE}_{\mathbf{A}_{s(k)}}$. Observe that $\text{MSE}_{\mathbf{A}_{s(k)}}$ corresponds to the optimal solution of $\mathcal{MP}_{\text{base}}(k)$. Because the optimal value of a mixed integer programming model minimizing the objective function is lower bounded by that of its LP-relaxation, we have $\text{MSE}_{\mathbf{A}_{s(k)}} \geq \text{MSE}_{\text{LB}}$.

Now let us derive a lower bound of $(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1}$. We start with noting that for an arbitrary j and a subset $s(k)$ that includes j , there exists $j_0 \in (\mathcal{J} \cup \mathcal{J}^l) \setminus \{j\}$ where $\{j_0\} \subset s(k) \setminus \{j\}$. Thus, by Lemma 1 we can realize the inequalities $R^2(\mathbf{A}_{s(k) \setminus \{j\}}, \mathbf{A}_j) \geq R^2(\mathbf{A}_{j_0}, \mathbf{A}_j) \geq \min_{j' \in (\mathcal{J} \cup \mathcal{J}^l) \setminus \{j\}} R^2(\mathbf{A}_{j'}, \mathbf{A}_j) = R^2_{-j}$.

Then, by Definition 1 and the fact that $0 \leq R^2(\mathbf{A}, \mathbf{b}) \leq 1$ for any \mathbf{A} and \mathbf{b} , we have

$$(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1} = \frac{1}{1 - R^2(\mathbf{A}_{s(k) \setminus \{j\}}, \mathbf{A}_j)} \geq \frac{1}{1 - R^2_{-j}}.$$

Finally, combining the lower bounds of $\text{MSE}_{\mathbf{A}_{s(k)}}$ and $(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1}$, the lowerbound of $s(\mathbf{A}_{s(k)})_j$ is given by

$$s(\mathbf{A}_{s(k)})_j = \sqrt{\text{MSE}_{\mathbf{A}_{s(k)}}(\mathbf{A}'_{s(k)}\mathbf{A}_{s(k)})_j^{-1}} \geq \sqrt{\text{MSE}_{\text{LB}}(k)} \sqrt{\frac{1}{1 - R^2_{-j}}}. \quad \square$$

Setting $s_j(k)_{\text{LB}} = \sqrt{\frac{\text{MSE}_{\text{LB}}(k)}{1 - R^2_{-j}}}$, we can formulate the relaxed constraint of (6) as

$$x_j^+ + x_j^- \geq t_{1-\frac{\alpha}{2}, n-k-1} s_j(k)_{\text{LB}} z_j, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l. \quad (7)$$

It is worthy to note that calculating $s_j(k)_{\text{LB}}$ is computationally efficient. The quantity $\text{MSE}_{\text{LB}}(k)$ is obtained by solving an LP and the coefficient of determination R^2_{-j} can be quickly obtained by the closed-form formula for $(m-1)$ simple linear regression models.

Because $s_j(k)_{\text{LB}}$ is a calculated constant and $t_{1-\frac{\alpha}{2}, n-k-1}$ is determined by the user, constraint (7) is a linear constraint. However, constraint (7) does not work properly with (5). This is because, given a solution to (5), both x_j^+ and x_j^- can be increased to satisfy (7) without violating the other constraints in (5) or changing the objective function value of the solution. To handle this issue, we introduce new variables z_j^+ and z_j^- , where $z_j^+ = 1$ if $x_j^+ > 0$, $z_j^+ = 0$ if $x_j^+ = 0$ and $z_j^- = 1$ if $x_j^- > 0$, $z_j^- = 0$ if $x_j^- = 0$. By replacing z_j with z_j^+ and z_j^- , (5) can be modified as follows:

$$\text{minimize} \quad \sum_{i \in \mathcal{I}} (e_i^{+2} + e_i^{-2}) \quad (8a)$$

$$\text{subject to} \quad e_i^+ - e_i^- = \sum_{j \in \mathcal{J} \cup \mathcal{J}^l} \tilde{a}_{ij} (x_j^+ - x_j^-) - b_i \quad \forall i \in \mathcal{I}, \quad (8b)$$

$$x_j^+ \leq M z_j^+, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l, \quad (8c)$$

$$x_j^- \leq M z_j^-, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l, \quad (8d)$$

$$\sum_{j \in \mathcal{J} \cup \mathcal{J}^l} (z_j^+ + z_j^-) = k, \quad (8e)$$

$$z_j^+ + z_j^- + z_{j^l}^+ + z_{j^l}^- \leq 1, \quad \forall j \in \mathcal{J}, \quad j^l = j + m, \quad (8f)$$

$$x_j^+ + x_j^- \geq t_{1-\frac{\alpha}{2}, n-k-1} s_j(k)_{\text{LB}} (z_j^+ + z_j^-), \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l, \quad (8g)$$

$$x_j^+, x_j^- \geq 0, \quad z_j^+, z_j^- \in \{0, 1\}, \quad \forall j \in \mathcal{J} \cup \mathcal{J}^l, \quad (8h)$$

$$e_i^+, e_i^- \geq 0, \quad \forall i \in \mathcal{I}. \quad (8i)$$

Note that z_j is replaced by z_j^+ or z_j^- in (8c) and (8d) and $z_j^+ + z_j^-$ is used instead of z_j in (8e) and (8f). Note that (8g) is added for the t -test, but it is a relaxed version of the exact t -test constraint (6). This relaxed constraint is useful in reducing the search space of the branch-and-bound algorithm of the solver, while the exact constraint is enforced by the technique presented in the subsequent section.

3.4 Final algorithm

Recall that constraint (8g) is a relaxed constraint derived from the lower bound of $s(A_{s(k)})$ and that a feasible solution for (8) can have statistically insignificant coefficients, while we want all coefficients to be statistically significant. Furthermore, it is challenging to formulate tests to check the residual assumptions in Section 2.3 as linear or convex constraints. To overcome these limitations, we propose a lazy constraint-based approach to solve the following problem.

$$\begin{aligned} & \text{minimize} && (8a) \\ & \text{subject to} && (8b)-(8i), \\ & && \text{coefficients } t\text{-tests constraints (2) from Section 2.2,} \\ & && \text{residual diagnostics constraints from Section 2.3.} \end{aligned} \quad (9)$$

Lazy callback is mainly employed in practical implementations using an optimization solver when the number of constraints in an MIP model is extremely large [13]. For example, we may use lazy callback when solving the traveling salesman problem (TSP) because it has a large number of subtour elimination constraints. Instead of adding all subtour elimination constraints at the root node of the branch-and-bound algorithm, we can iteratively and selectively add some of these constraints. In detail, when the solver arrives at a branch-and-bound node and the solution contains a subtour, lazy callback allows us to add the corresponding subtour elimination constraint at the current node. Although our MIQP model does not have an extremely large number of constraints, we can use lazy callback to stop at a branch-and-bound node and check if the solution completely satisfies all of the tests and diagnostics. Recently, Bertsimas and Li [2] also use the lazy constraint to overcome difficulties related to the coefficient t -test. Our lazy constraint-based procedure, conducted independently of Bertsimas and Li [2], is provided in Algorithm 1.

In the algorithm, J_{node} is the index set of selected variables at the current node in the brand-and-bound tree. The lazy constraint $\sum_{j \in J_{\text{node}}} (z_j^+ + z_j^-) \leq k-1$ is added to the model when (i) a linear model

Algorithm 1 Lazy constraint procedure (at each node J_{node})

Input: a set of α (significance levels for each statistical test)

```
1: for each  $j' \in J_{\text{node}}$  do
2:   if the coefficient corresponding to variable  $j'$  is not statistically significant then
3:     add a lazy constraint  $\sum_{j \in J_{\text{node}}} (z_j^+ + z_j^-) \leq k - 1$  to the model
4:   return
5:   end if
6: end for
7: if the model with variables in  $J_{\text{node}}$  fails to pass the residual diagnostics test then
8:   add a lazy constraint  $\sum_{j \in J_{\text{node}}} (z_j^+ + z_j^-) \leq k - 1$  to the model
9: end if
10: return
```

associated with $j \in J_{\text{node}}$ fails to pass at least one of the statistical tests for the coefficients or (ii) the model fails to pass the residual diagnostics test. The lazy constraint eliminates the solution of selecting all explanatory variables in J_{node} from the feasible region. For example, let us consider a problem with 10 explanatory variables with the associated binary variables set $\{z_1, \dots, z_{10}\}$. We assume that we set $k = 4$ and the solver is currently at the node with a solution having $z_1^+ = z_4^+ = z_9^- = z_{10}^+ = 1$ and $z_j^+, z_j^- = 0$ for all others. The procedure generates a linear model with the selected explanatory variable set $J_{\text{node}} = \{1, 4, 9, 10\}$ and conducts statistical tests and diagnostics for the model. When the regression model fails to pass at least one of the tests, lazy constraint $\sum_{j \in J_{\text{node}}} (z_j^+ + z_j^-) = z_1^+ + z_4^+ + z_9^- + z_{10}^+ \leq 3$ is added to the MIQP model. As a result, the solution $z_1^+ = z_4^+ = z_9^- = z_{10}^+ = 1$ (the others are zero due to constraint (8e)) becomes infeasible in all of the subsequent branches.

When there is no feasible solution with a fixed k value, the user can relax the cardinality constraint and consider different k values. This search can easily be performed by re-running the algorithm with different k values; we do not consider multiple k values in the proposed alternative solution procedure.

Table 1 summarizes the model building and validation procedures presented in this section, along with the associated constraints and sections. Finally, we remark that the proposed lazy constraint framework can easily be extended and incorporate other tests such as F -tests or pairwise multicollinearity tests based on user needs.

Table 1: Constraints for model building and validation procedures

Model building and validation procedure	Associated constraints	Section reference
Nonlinear variable transformations	(8f)	Section 2.1
Significance of estimated coefficients	(8g), lazy constraint	Section 2.2
Residual tests	lazy constraint	Section 2.3

4 Alternative solution procedure for infeasible problems

Based on the algorithm in Section 3.4, we can obtain the best subset satisfying all tests for model validation discussed in Section 2. However, when all possible subsets violate at least one of the tests and assumptions, the model (8) becomes infeasible and the algorithm will not return a solution. Even in this case, a near-feasible solution is desired (i.e., satisfying most of the tests and assumptions while mildly violating a few of the tests). To find such solutions, we propose an algorithm, referred to as *alternative solution procedure*. The algorithm searches for a near-feasible solution based on the model development strategy in Figure 1 and penalties for violation of regression tests and assumptions.

The alternative solution procedure is invoked at each branch-and-bound node while no feasible solution satisfying all tests and diagnostics has been found by the algorithm. Once a feasible solution is found, the alternative solution procedure will not be invoked because at least one feasible solution to the problem satisfying all of the tests and diagnostics exists. When the alternative solution procedure is invoked, it compares a *new subset* with the incumbent *best subset* (not feasible, but the best alternative near-feasible subset) and decides which subset is better. Our model updates and keeps only one alter-

native solution until it finds a feasible solution. The new subset, denoted as S_{new} , is the subset at the current branch-and-bound node and the incumbent best subset, denoted as S_{best} , is the best alternative subset found so far. Remark that S_{new} and S_{best} must be infeasible solutions with some insignificant coefficients or residual assumption violations because the alternative solution procedure keeps S_{best} until a feasible solution is found. To simplify the discussion, we will use S_{best} and S_{new} to refer to both selected subsets and linear regression models.

Note that one of the simplest approaches to comparing S_{new} , S_{best} , and all other subsets is to add penalty terms to the objective function based on diagnostic violations such as the average p -value or the number of insignificant variables. Before presenting the details of the alternative solution procedure, we first explain why the simple penalty approach may fail and why the proposed procedure is needed. First, the penalty approach cannot effectively reflect the overall development procedure of a linear regression model. In typical model development, explained in Section 2, we first establish a linear regression model with statistically significant coefficients. We then check the residual assumptions via diagnostics. That is, we should consider the residual assumptions after establishing a statistically significant model. This cannot be achieved with the simple penalty approach. Next, there exist non-trivial cases where the penalty approach may fail to select the better subset. This problem is discussed with the illustrative examples presented in Table 2.

Table 2: Illustrative examples for the alternative solution procedure

Cases	Set of p -values		Better set
Case 1	S_{best}	$\{0.06, 0.06, 0.06, 0.025, 0.02\}$	S_{new}
	S_{new}	$\{0.055, 0.055, 0.04, 0.025, 0.02\}$	
Case 2	S_{best}	$\{0.06, 0.06, 0.06, 0.025, 0.02\}$	S_{best}
	S_{new}	$\{0.95, 0.04, 0.04, 0.025, 0.02\}$	
Case 3	S_{best}	$\{0.06, 0.06, 0.06, 0.025, 0.02\}$	S_{new}
	S_{new}	$\{0.065, 0.04, 0.04, 0.025, 0.02\}$	

Table 2 presents three cases of determining which solution is better between S_{best} and S_{new} , where k and α are set to 5 and 0.05, respectively. The set of p -values from the statistical test for the coefficients is shown for each case. In Case 1, it is not surprising that we choose S_{new} because the number of insignificant coefficients in S_{new} is less than that in S_{best} (i.e., $3 > 2$), and the average of the violating p -values for S_{new} is also less than that for S_{best} (i.e., $0.060 > 0.055$). On the other hand, S_{best} must be better in Case 2 because the average of the violating p -values for S_{new} is much larger than that for S_{best} ($0.06 < 0.95$), even though the number of insignificant coefficients in S_{new} is greater than that in S_{best} ($3 > 1$). In Case 3, although the average of the violating p -values for S_{best} is less than that for S_{new} ($0.60 < 0.65$), selecting S_{new} as the better solution seems reasonable because the difference is negligible while the number of insignificant coefficients in S_{best} is significantly greater than in S_{new} (i.e., $3 > 1$). These illustrative examples indicate that we should consider both the average of the violating p -values and the number of insignificant coefficients. However, due to the different magnitudes of the measures, it is not trivial to set appropriate weights. Consequently, it is necessary to develop an intuitive procedure that reflects the framework of linear regression model construction and further alleviates the scale difference for measures of diagnostic violation. The experiment in Section 5 shows that the alternative solution procedure outperforms a simple penalty approach.

Now, we discuss the alternative solution procedure. To explain this more efficiently, we first define the functions related to the significance test for the coefficients.

- $\pi(S)$: number of insignificant coefficients in the model with subset S
- $E(S)$: average p -values of the insignificant coefficients in the model with subset S ; 0 if all coefficients in subset S are statistically significant
- $r_l(S)$: p -value from the residual linearity test for subset S
- $r_h(S)$: p -value for the residual heteroscedasticity tests, which is the maximum p -value between the p -values of the absolute residual test and the Breusch-Pagan test
- $f_q(S_1, S_2)$: *decision by quality*; S_1 and S_2 are subsets

$f_t(S_1, S_2; \tau)$: *decision with tolerance*; S_1 and S_2 are subsets

The last two functions, $f_q(S_1, S_2)$ and $f_t(S_1, S_2; \tau)$, decide whether S_1 or S_2 is better based on different principles, which will be discussed in detail later in this section.

The overall alternative solution procedure is presented in Algorithm 2. Step 1 is for when S_{new} is the statistically significant model while S_{best} is not. Hence, it is natural to return S_{new} . Step 2 considers the opposite case to that of Step 1. In Steps 3-4, non-trivial cases are considered: both models are statistically significant in Step 3 and both models are statistically insignificant in Step 4. For these two cases, f_q , referred to as decision by quality, and f_t , referred to as decision by tolerance, are employed to make a decision.

Algorithm 2 Alternative solution procedure

Input: $S_{\text{best}}, S_{\text{new}}$

Output: the better subset between S_{best} and S_{new}

- 1: **if** $\pi(S_{\text{best}}) > 0$ **and** $\pi(S_{\text{new}}) = 0$ **then return** S_{new}
 - 2: **else if** $\pi(S_{\text{best}}) = 0$ **and** $\pi(S_{\text{new}}) > 0$ **then return** S_{best}
 - 3: **else if** $\pi(S_{\text{best}}) = 0$ **and** $\pi(S_{\text{new}}) = 0$ **then return** $f_q(S_{\text{best}}, S_{\text{new}})$
 - 4: **else if** $\pi(S_{\text{best}}) > 0$ **and** $\pi(S_{\text{new}}) > 0$ **then return** $f_t(S_{\text{best}}, S_{\text{new}}; \tau)$
 - 5: **end if**
-

Function $f_q(S_{\text{best}}, S_{\text{new}})$ returns the better solution between S_{best} and S_{new} by quantifying the quality of the solutions using the penalty function q for solution S , introduced below.

- MSE(S) : the MSE of solution S
- $\lambda_\pi, \lambda_E, \lambda_l, \lambda_h$: penalty parameters for π , E , r_l , and r_h , respectively
- $w_1(p, \alpha)$: percentage transform function for $E(S)$; $\frac{\max(p - (1 - \alpha), 0)}{\alpha}$
- $w_2(p, \alpha)$: percentage transform function for $r_l(S)$ and $r_h(S)$; $\frac{\max((1 - \alpha) - p, 0)}{1 - \alpha}$

$$q(S; \lambda_\pi, \lambda_E, \lambda_l, \lambda_h) = \text{MSE}(S) + \lambda_\pi \pi(S) + \lambda_E w_1(E(S), \alpha_E) + \lambda_l w_2(r_l(S), \alpha_l) + \lambda_h w_2(r_h(S), \alpha_h)$$

Note that function q includes percentage transform functions $w_1(p, \alpha)$ and $w_2(p, \alpha)$ to indicate the percentage gap between the significance levels. We introduce these functions because first we want to accurately measure the insignificance of the p -values when the significance levels are at different scales, and second, we do not want to apply penalties if the p -value falls within the significance level.

Two examples are provided for w_1 and w_2 to support our claims and demonstrate their necessity. The first example is of penalty nullification. Suppose $r_h(S_1) = 0.2$ and $r_h(S_2) = 0.04$, which implies the residuals of S_1 are consistent while those of S_2 have heteroscedasticity. Given $\alpha_h = 0.9$, we can calculate $w_2(r_h(S_1), \alpha_h) = \frac{\max((1 - 0.9) - 0.2, 0)}{0.1} = 0$, and $w_2(r_h(S_2), \alpha_h) = \frac{\max((1 - 0.9) - 0.04, 0)}{0.1} = \frac{0.06}{0.1} = 0.6$. The penalties make sense because S_1 does not violate the test. Hence, the evaluation of S_1 is based only on the MSE and the significance of the coefficients. The second example demonstrates the role of scaling between p -values from different statistical tests. Suppose that we get $E(S_3) = 0.15$, $r_h(S_3) = 0.05$. Further, α_E and α_h are both set to 0.9. Then, S_3 violates the t -tests of the coefficients by $0.15 - (1 - 0.9) = 0.05$, which is equal to the value of the heteroscedasticity test violation $(1 - 0.9) - 0.05 = 0.05$. However, the violation of 0.05 is relatively small for the possible t -test violation range $[0, 0.9]$ compared to the possible residual test violation range $[0, 0.1]$. Thus, despite the same violation size, the significance of the violation can differ depending on the tests and diagnostics. We can scale the magnitude of the violations through w_1 and w_2 : $w_1(E(S_3), \alpha_E) = \frac{0.15 - 0.1}{0.9} \approx 0.056$, and $w_2(r_h(S_3), \alpha_h) = \frac{0.1 - 0.05}{0.1} = 0.5$.

We now discuss the *decision with tolerance* presented in Algorithm 3. Step 1 is key to our procedure. If $E(S_{\text{new}}) - E(S_{\text{best}})$ exceeds the tolerance τ , a positive parameter predetermined by the user, then the alternative solution procedure concludes that S_{best} is better. When the average of the violating p -values of S_{new} is smaller or not significantly worse (smaller than τ) than that of S_{best} , we conclude S_{new} is competitive and investigate further in Steps 3-7 by comparing the two solutions based on two factors, π and E . Step 3 concludes that, if S_{new} is smaller than S_{best} in terms of these two factors, S_{new} is better. The opposite case is shown in Step 4. If there is a conflict between the two factors, f_q settles the decision.

Note that Algorithms 2 and 3 imitate the linear regression model building procedure described in Section 2 by using statistical significance tests first to make the decision, followed by other diagnostics.

Algorithm 3 *Decision with tolerance, f_t*

Input: $S_{\text{best}}, S_{\text{new}}, \tau$

Output: the better subset between S_{best} and S_{new}

```

1: if  $E(S_{\text{new}}) - E(S_{\text{best}}) > \tau$  then return  $S_{\text{best}}$ 
2: else
3:   if  $E(S_{\text{best}}) > E(S_{\text{new}})$  and  $\pi(S_{\text{best}}) > \pi(S_{\text{new}})$  then return  $S_{\text{new}}$ 
4:   else if  $E(S_{\text{best}}) < E(S_{\text{new}})$  and  $\pi(S_{\text{best}}) < \pi(S_{\text{new}})$  then return  $S_{\text{best}}$ 
5:   else return  $f_q(S_{\text{best}}, S_{\text{new}})$ 
6:   end if
7: end if

```

We illustrate the entire alternative solution procedure using the three toy examples presented in Table 2. Suppose that our model is searching a subset whose cardinality k is set to 5 and that a feasible solution has not yet been found. A user sets the significance levels and τ to 95 percent and 0.1, respectively.

- **Case 1.** Our model cannot determine which solution is better in the first step because $\pi(S_{\text{best}}) = 3 > 0$ and also $\pi(S_{\text{new}}) = 2 > 0$. Hence, decision with tolerance is invoked (Algorithm 3). In Step 1, the algorithm computes $E(S_{\text{best}}) = \frac{1}{3}(0.06 + 0.06 + 0.06) = 0.06$, $E(S_{\text{new}}) = \frac{1}{2}(0.055 + 0.055) = 0.055$, and $E(S_{\text{new}}) - E(S_{\text{best}}) = -0.005 < \tau = 0.1$. Thus, the next steps compare $E(S_{\text{new}})$ with $E(S_{\text{best}})$ and $\pi(S_{\text{new}})$ with $\pi(S_{\text{best}})$. Since $E(S_{\text{new}}) < E(S_{\text{best}})$ and $\pi(S_{\text{new}}) < \pi(S_{\text{best}})$, S_{new} is determined to be the better solution.
- **Case 2.** Because $\pi(S_{\text{best}})$ and $\pi(S_{\text{new}})$ are both greater than 0, our model cannot determine which solution is better in Step 4 of Algorithm 2, and f_t is called. In Algorithm 3, since $E(S_{\text{new}}) - E(S_{\text{best}}) = 0.95 - 0.06 = 0.89 > \tau = 0.1$ in Step 1, S_{best} is returned.
- **Case 3.** Similar to the previous cases, f_t is called in Step 4 of Algorithm 2. In Algorithm 3, although $E(S_{\text{new}}) - E(S_{\text{best}}) = 0.065 - 0.06 = 0.005$ and $E(S_{\text{new}})$ is greater than $E(S_{\text{best}})$, Steps 3-6 are considered because $E(S_{\text{new}}) - E(S_{\text{best}}) = 0.005 < 0.1 = \tau$. However, since $E(S_{\text{new}})$ is greater than $E(S_{\text{best}})$ ($0.065 > 0.06$) and $\pi(S_{\text{new}})$ is less than $\pi(S_{\text{best}})$ ($1 < 3$), the winning solution is determined using f_q .

Overall, we find that the alternative solution procedure selects the desired solution discussed in Table 2. During the alternative solution procedure, we first compare S_{new} with S_{best} using the significance of the estimated coefficients, in particular with $\pi(S)$ and $E(S)$. If this step cannot lead to a decision, residual diagnostics are considered. This multistage procedure reflects the linear regression model building framework discussed in Section 2. The procedure also deals with the possible issues illustrated in Case 2 and 3 in Table 2 by introducing f_t . A user can intuitively provide a value of τ based on the average difference in the p -value that they can permit. In Experiment 4 of Section 5, we provide experimental results that illustrate the benefit of using the proposed procedure rather than a simple penalty function.

In this alternative solution framework, we empirically adopt $E(S)$ and $\pi(S)$ as measures evaluating how statistically significant a regression model is. The practical examples discussed above justify the measures. We finally note that these measures can be varied or alternatives can be employed to reflect their own perspectives on the statistical significance. In the next sections, $\mathcal{MP}_{\text{lazy}}(k)$ denotes our final algorithm: solving (8) with Algorithm 1 and the alternative solution procedure in Algorithms 2 and 3.

Finally, we summarize the proposed approach as follows. Our ultimate goal is solving the MIP model (9). Due to the nonlinearity of the exact t -test and diagnostic constraints, we instead start with solving the MIP model (8). While solving the model, a lazy constraint for the t -test and diagnostic is added at each branch-and-bound node if needed. Simultaneously, we keep the best near-feasible solution found by the alternative solution procedure invoked at the node. Once a feasible solution is found during the search, the alternative solution procedure is not invoked. If we do not find a feasible solution until the solution search terminates, the model outputs the best near-feasible solution found by the alternative solution procedure.

5 Computational Results

In this section, the results of numerical experiments using the proposed model and benchmarks are presented. The experiments are designed to demonstrate the performance of the proposed model and the necessity for the alternative solution procedure.

5.1 Benchmark dataset description and preprocessing

Twelve publicly available datasets are used in the experiment. We collect five datasets for regression from the UCI machine learning data repository [23] and seven datasets from other sources. The features of the experimental datasets are summarized in Table 3.

Table 3: Datasets used in the experimental demonstration

Dataset	Observations (n)	Explanatory variables		Source
		Raw	Preprocessed (m)	
Housing	506	13	26	UCI data repository
Servo	167	4	38	UCI data repository
AutoMPG	392	8	50	UCI data repository
Automobiles	159	26	124	UCI data repository
Winequality	1599	11	22	UCI data repository
Bodyfat	252	15	30	Johnson [17]
Barro	161	13	26	Koenker and Machado [20]
Carseats	400	10	28	James et al. [16]
Crime	630	22	50	Cornwell and Trumbull [9]
Framing	265	14	30	Brader et al. [5]
Griliches	758	19	50	Blackburn and Neumark [4]
Hprice3	321	19	22	Wooldridge [37]

We preprocess the datasets as follows. First, we remove records which include at least one missing value. Second, dummy variables are introduced for the categorical variables. Third, nominal variables are changed into numerical variables. Next, log-transformed variables are created for the numerical variables in the original data. Finally, all of the datasets are standardized.

5.2 Experimental design

In this section, we present the design of the five experiments. To demonstrate the practical viability of our model, we set a time limit of 600 seconds for every experiment, except for Experiment 2 where the time limit is set to 3600 seconds to compare the performances when the algorithms have sufficient amount of time. The algorithm time includes the computation time for big- M in (5c) and $s(k)_{LB}$ in (7). We set several parameters for the experiments: $\alpha_E = 0.95$ and $\alpha_l = \alpha_h = 0.99$, where α_E , α_l , and α_h are the significance levels for the t -tests for the coefficients, diagnostics for residual linearity, and residual heteroscedasticity, respectively. Parameters for the alternative solution procedure is set to as follows: $\tau = 0.1$; $\lambda_E = 1.5$; $\lambda_\pi = \lambda_h = \lambda_l = 0.125$. The parameters are tuned to balance the fitting errors and validation measures based on the datasets used in our experiment. In our pilot computational experiments, we found that the suggested parameters return good results for various datasets. Therefore, for a new dataset, we recommend the user to start with the suggested parameters and iteratively adjust the parameters to improve the specific validation measures π , E , r_l , and r_h as needed. Alternatively, one can choose the parameters by cross-validation. In general, increasing the penalty parameter improves the corresponding validation measure. In Experiment 5, we show how the validation measures change as we change the penalty parameters and we hope this will help the user to successfully search for good parameters and determine their own best parameters.

Experiment 1: Proposed model vs. simple benchmark models We compare $\mathcal{MP}_{\text{lazy}}(k)$ with three benchmark models: $\mathcal{MP}_{\text{base}}(k)$, $\mathcal{FS}(k)$ (a forward selection algorithm), and LASSO. With this experiment, the performance of the proposed model in terms of the aforementioned statistical significance and regression assumptions can be verified.

The forward selection algorithm iteratively adds an explanatory variable until k variables are selected while the original and log-transformed variables cannot be in the subset simultaneously. The algorithm is constructed by modifying a standard forward selection algorithm to select from among variables and their log-transforms. Specifically, once a variable is selected, both the original and transformed variables are excluded from the candidate set of variables. If $j^* \in J$, then the log-transformed variable is excluded as well. If $j^* \in J^l$, then the original variable is excluded as well. Thus, $\mathcal{FS}(k)$ always provides a solution, with at most one variable from an original and transformed variable pair.

LASSO is a very popular regularized regression method that penalizes the L_1 norm of coefficients. The key advantage of LASSO is that it automatically performs variable selection while the coefficients shrink through the regularization. LASSO and the proposed model are different in that our model has an L_0 norm-based constraint (penalty) and diagnostic constraints. Intrinsically, it is not straightforward to include the additional constraints in LASSO.

For each dataset, all models and algorithms are tested over $k = 3, 4, \dots, 10$. Because LASSO does not select k explicitly, the proper penalty parameters of LASSO that give $k = 3, 4, \dots, 10$ are found by grid search. For each case, we check the goodness-of-fit with an adjusted R^2 , denoted as R_{adj}^2 . Although the objective of all models and algorithms is to maximize the SSE, this is equivalent to maximizing R_{adj}^2 because k is fixed. Also, because R_{adj}^2 ranges from 0 to 1, we can easily compare the goodness-of-fit over different k values and datasets.

Experiment 2: Proposed model vs. iterative model We compare our model with the iterative algorithm used in Bertsimas and King [1] for cases where the solution for $\mathcal{MP}_{\text{base}}(k)$ violates some of the tests and diagnostics. The algorithm iteratively adds constraints to avoid subsets with insignificant coefficients. We will refer to the iterative model as $\mathcal{MP}_{\text{iter}}(k)$ in this paper. Recall that a key feature of our algorithm is the ability to avoid iterative calls of the MIQP solver, where iterative approaches solve multiple MIQPs by adding constraints. In this comparison, we demonstrate the effectiveness of our lazy constraint-based algorithm. As the two models have different constraints and parameters, we compare the two models using common constraints, the number of variables selected and t -tests. That is, we do not consider the residual tests in this experiment.

Experiment 3: Proposed model vs. model without the relaxed t -test constraints We show the effectiveness of the relaxed t -test constraints (8g) derived in Section 3.3 by comparing our model with and without the relaxed constraint (8g). The experiment is indeed equivalent to the comparison between our model and the recently proposed model by Bertsimas and Li [2] without multicollinearity constraints. We evaluate the models based on two criteria: (i) the solution time and (ii) the number of branch-and-bound nodes searched until the optimal solution is obtained. If the model with (8g) is faster and searches less number of nodes, then we can show that (8g) is effective. In this experiment, we use the cases where our proposed model is able to find the optimal solution within a given time limit in the previous experiments.

Experiment 4: Alternative solution procedure vs. simple penalty function We demonstrate the effectiveness of the alternative solution procedure by comparing it with a simple penalty function. The benchmark is obtained by replacing the alternative solution procedure with the simple penalty function f_q . The benchmark model will be referred to as $\mathcal{MP}_{\text{penalty}}(k)$.

Experiment 5: Sensitivity analysis In this experiment, we demonstrate how the solutions obtained from $\mathcal{MP}_{\text{lazy}}$ differ and lead to different validation measures when the penalty parameters $\lambda_E, \lambda_\pi, \lambda_h$, and λ_l change. The Bodyfat dataset is used. We set $k = 7, 8, 9$, and 10, excluding $k = 3, 4, 5$, and 6 as our model $\mathcal{MP}_{\text{lazy}}$ found feasible solutions (see the results of Experiment 1 in Section 5.3) and the penalty parameters did not affect the results in these cases because they are used to search for alternative solutions when feasible solutions are not available.

To check the effect of λ_E , we run the algorithm for $\lambda_E \in \{0.5, 1, 1.5, \dots, 4.5, 5\}$ with the other parameters fixed using the values presented in Section 5.2. Similarly, to check the effect of λ_π , we run the algorithm for $\lambda_\pi \in \{0.025, 0.05, \dots, 0.225, 0.25\}$ with the other parameters fixed. Because $E \in [0, 1]$ and $\pi \in \{1, 2, \dots, k\}$ are on different scales, different values are checked for λ_E and λ_π in this experiment. Note that, of the penalty parameters, we only vary λ_π and λ_E . This is because the solutions from the experiment for Bodyfat do not exhibit heteroscedasticity (see the results for Bodyfat in Figure 4). Because the p -values are either very large or small, changing the associated penalty parameters λ_h and λ_l does not change r_h and r_l in this experiment.

5.3 Experimental results

We now present the experimental results. For the numerical experiments, we utilize Intel(R) Core(TM) i7-8700 CPU @ 3.40GHz (8 CPUs) and 32GB RAM. All models and algorithms are implemented with Python, in which mathematical models are solved using Gurobi 9.0.0. For the construction of the linear model in $\mathcal{FS}(k)$, we employ the Python package *statsmodels* [32]. In every experiment, none of the solutions from the comparative models demonstrate linearity between their residuals and fitted values, so the corresponding results are not provided.

Results for Experiment 1: Proposed model vs. simple benchmark models To measure the explanatory power of each model, we introduce a measure for relative explanatory power $REP = \frac{R_{adj}^2(S)}{R_{adj}^2(S_{\mathcal{MP}_{base}})}$, where $S_{\mathcal{MP}_{base}}$ is a subset obtained by solving \mathcal{MP}_{base} and S is the solution of the corresponding model. We use the R_{adj}^2 of \mathcal{MP}_{base} as a denominator of REP since \mathcal{MP}_{base} provides the greatest R_{adj}^2 of the compared models (except for LASSO) because it does not have any diagnostic constraints. Note that REP of LASSO can be greater than 1, because LASSO does not include the log transform constraint (5e).

We present the summarized results for the datasets and the number of variables selected (k) in Tables 4 and 5, respectively. The complete results are available in the online supplement. In both tables, REP is the average over the dataset or the k values. To test the performance when using only feasible cases, we also present REP_{feas} , which only considers cases with feasible solutions available when calculating the average. Additionally, we count the number of cases satisfying (i) the t -test and residual test, (ii) the t -test only, and (iii) the residual test only.

Table 4: Results by dataset

	REP				REP_{feas}				Execution time			
Dataset	\mathcal{MP}_{lazy}	\mathcal{MP}_{base}	\mathcal{FS}	LASSO	\mathcal{MP}_{lazy}	\mathcal{MP}_{base}	\mathcal{FS}	LASSO	\mathcal{MP}_{lazy}	\mathcal{MP}_{base}	\mathcal{FS}	LASSO
Housing	0.950	1.000	1.000	0.979	0.950	1.000	1.000	0.979	329.676	7.802	0.066	0.001
Servo	0.921	1.000	1.000	0.946	0.367	1.000	1.000	0.915	494.756	60.021	0.094	0.000
AutoMPG	0.929	1.000	0.998	0.927	0.929	1.000	0.998	0.927	478.818	332.581	0.140	0.001
Automobiles	0.992	1.000	1.000	0.943	0.992	1.000	1.000	0.943	549.824	513.232	0.348	0.001
Winequality	0.872	1.000	1.000	0.997	0.658	1.000	1.000	1.000	487.790	41.402	0.058	0.008
Bodyfat	0.783	1.000	1.000	1.002	0.816	1.000	1.000	1.002	469.123	22.906	0.062	0.004
Barro	0.910	1.000	0.989	0.833	1.000	1.000	0.974	0.593	376.991	3.597	0.056	0.001
Carseats	0.983	1.000	1.000	0.948	1.000	1.000	1.000	0.929	155.391	4.808	0.071	0.001
Crime	0.847	1.000	0.993	0.887	0.823	1.000	0.992	0.872	420.495	23.823	0.141	0.003
Framing	0.999	1.000	1.000	0.997	1.000	1.000	1.000	1.000	303.066	7.046	0.069	0.001
Griliches	0.987	1.000	0.997	0.900	0.987	1.000	0.997	0.900	163.162	184.169	0.154	0.001
Hprice3	0.950	1.000	0.988	0.961	0.921	1.000	0.989	0.941	335.505	3.551	0.047	0.001
	t -test & residual test				t -test				Residual test			
Dataset	\mathcal{MP}_{lazy}	\mathcal{MP}_{base}	\mathcal{FS}	LASSO	\mathcal{MP}_{lazy}	\mathcal{MP}_{base}	\mathcal{FS}	LASSO	\mathcal{MP}_{lazy}	\mathcal{MP}_{base}	\mathcal{FS}	LASSO
Housing	8	1	1	1	8	8	8	5	8	1	1	1
Servo	1	0	0	0	8	7	7	8	1	0	0	0
AutoMPG	8	0	0	0	8	8	8	3	8	0	0	0
Automobiles	8	3	5	0	8	8	8	1	8	3	5	0
Winequality	3	0	0	0	7	5	5	4	3	0	0	0
Bodyfat	4	0	0	0	4	0	0	1	8	0	0	0
Barro	3	3	3	0	3	3	3	0	8	8	8	8
Carseats	6	5	5	7	6	5	5	7	8	8	8	8
Crime	7	0	0	1	8	8	8	1	7	0	0	2
Framing	4	4	4	5	4	4	4	5	8	8	8	8
Griliches	8	4	2	3	8	6	5	3	8	4	2	3
Hprice3	5	0	0	0	7	6	4	4	5	0	0	0

In Table 4, we observe that \mathcal{MP}_{lazy} obtains solutions with a REP of 90% or above except for the Winequality, Bodyfat, and Crime datasets. This indicates that our model maintains explanatory power while satisfying all diagnostics constraints. Column REP_{feas} also indicates that our model maintains an R_{adj}^2 closer to the base model. The results in the ' t -test & residual test' column indicate that our model is able to find substantially more linear models satisfying both the statistical significance of all coefficients and the residual assumptions for most of the datasets. In particular, our model finds optimal solutions satisfying all diagnostics constraints for all cases for four data sets (bolded in the ' t -test & residual test' column). On the other hand, our model finds the same number of feasible solutions as the benchmarks for the Framing dataset, with the results from our model for REP_{feas} equal to 1 for this

Table 5: Results by k

k	REP				REP_{feas}				Execution time			
	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO
3	0.952	1.000	0.996	0.902	0.947	1.000	0.996	0.899	30.671	14.843	0.050	0.001
4	0.869	1.000	0.997	0.913	0.869	1.000	0.997	0.913	114.242	34.682	0.065	0.001
5	0.917	1.000	0.995	0.926	0.909	1.000	0.995	0.922	294.359	64.270	0.083	0.002
6	0.913	1.000	0.999	0.959	0.916	1.000	0.999	0.960	415.163	95.284	0.102	0.002
7	0.924	1.000	0.998	0.956	0.937	1.000	0.997	0.945	517.561	128.043	0.118	0.002
8	0.932	1.000	0.998	0.972	0.939	1.000	0.998	0.962	522.185	131.130	0.136	0.002
9	0.921	1.000	0.998	0.978	0.915	1.000	0.999	0.964	578.240	141.099	0.149	0.003
10	0.948	1.000	0.998	0.977	0.958	1.000	0.998	0.954	603.944	142.587	0.166	0.003
t -test & residual test					t -test				Residual test			
k	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO
3	11	4	4	5	12	11	11	9	11	4	4	6
4	12	4	4	3	12	11	11	7	12	4	4	5
5	11	4	4	3	12	11	11	7	11	4	4	4
6	9	3	3	2	11	10	10	6	10	4	4	3
7	7	3	2	2	9	9	8	5	10	5	4	3
8	6	1	1	1	9	7	5	3	9	4	4	3
9	5	0	1	1	8	5	5	3	9	3	4	3
10	4	1	1	0	6	4	4	2	8	4	4	3

dataset. In fact, the corresponding solutions for $\mathcal{MP}_{\text{base}}$ are exactly the same as those of our model. This indicates that, although $\mathcal{MP}_{\text{base}}$ finds an optimal solution without considering any diagnostics or statistical significance, the solution fortunately has no insignificant coefficients and satisfies the residual assumptions. Thus, these results in Table 4 make clear that our model is able to provide a quality linear model independent of the dataset. Finally, the execution time indicates that our model can find linear models within a practical timeframe.

Table 5 presents the results by averaging or summing the number of variables selected (k). We can verify that the results of our model for REP_{feas} are close to 1, and the results in the ‘ t -test & residual test’ column are substantially better than those of the benchmarks, as in Table 4. This suggests that our model can also generate quality subsets regardless of the size of k .

In Figure 4, we check the results for the alternative solution procedure for the following cases where no feasible solution is found within the 600-second time limit: Servo, Winequality, Bodyfat, Barro, and Framing with $k = 7, 8, 9, 10$. In the plot matrix of Figure 4, the horizontal and vertical axes represent the datasets and the performance measures, respectively. Note that $E = 0$ if a solution has no statistically insignificant coefficient because, as illustrated in Section 4, a penalty term is activated when a violation of the corresponding statistical test occurs. Also, heteroscedasticity would not be a concern for the Barro and Framing dataset because every case in the dataset satisfies the regression assumption ($r_h > 0.01$).

The plots in Figure 4 explicitly illustrate the promising performance of the alternative solution procedure. As seen in the charts in the first row, the REP of the proposed model remains above 0.89 except for the Bodyfat dataset. Moreover, except for the two cases (Framing with $k = 7, 8$) including the case marked with a ‘*’, the measures π , E , and r_h for the alternative solutions are better than or equal to those of the benchmarks. These results indicate that the alternative solutions improve upon those of the benchmarks in most cases. For example, for Winequality ($k = 8$), the π and E of $\mathcal{MP}_{\text{lazy}}$ are considerably lower than those of the benchmarks while it has REP greater than 0.99. On the other hand, the REP for the alternative solutions for Bodyfat is relatively less than for the other results. However, π and E are significantly better than the benchmarks. This indicates that our model sacrifices explanatory power (i.e., MSE) to improve the statistical tests and diagnostics (π and E).

We investigate the case marked with a ‘*’ (Framing with $k = 8$) more in detail, in which the alternative solutions do not outperform the benchmark solutions for every measure. This is due to our algorithm’s tolerance parameter τ . Let S'_{current} be the solution of $\mathcal{MP}_{\text{lazy}}$ with $\pi = 3$ and $E = 0.1719$ and S'_{new} be the solution for LASSO with $\pi = 2$ and $E = 0.5577$ from the result in Figure 4. These two solutions can be compared using our alternative solution procedure. Based on Algorithms 2 and 3, f_t is called in Step 4 of Algorithm 2. Then Step 1 of Algorithm 3 compares $E(S'_{\text{new}})$ and $E(S'_{\text{current}})$ accounting for the given tolerance parameter $\tau = 0.1$. Because $E(S'_{\text{new}}) - E(S'_{\text{current}}) = 0.3858 > 0.1$, the alternative solution procedure concludes that S'_{current} is a better solution, which is a clearly reasonable decision. We

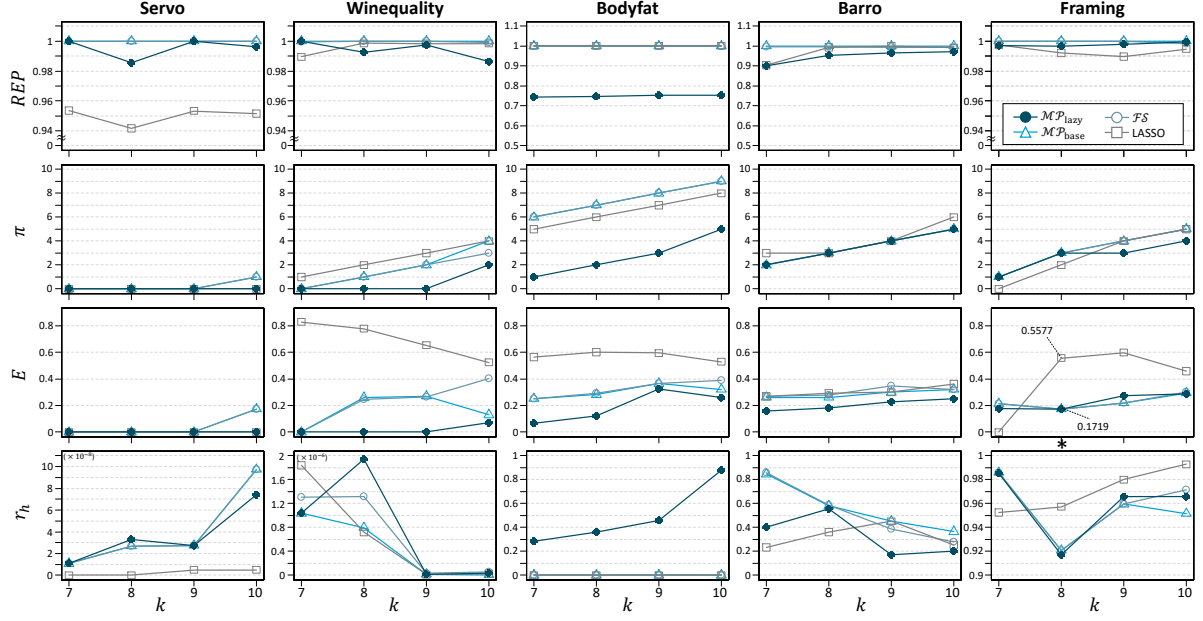


Figure 4: Comparison of the alternative and benchmark solutions. In each plot, the horizontal axis is the number of selected variables (k).

find that LASSO gives a better solution in only one case, Framing with $k = 7$.

Figure 5 presents representative residual plots (residuals versus fitted values) for the AutoMPG ($k = 8$), Automobile ($k = 3$), and Servo ($k = 4$). In Table 6, the corresponding p -values are presented. The plots in Figure 5 show that the variance of the residuals from the benchmarks gradually increases or decreases, while the linear regression model derived from our model has a relatively consistent residual trend in relation to the fitted values. The p -values in Table 6 show that our model provides better solutions because a higher p -value is preferable for the heteroscedasticity tests.

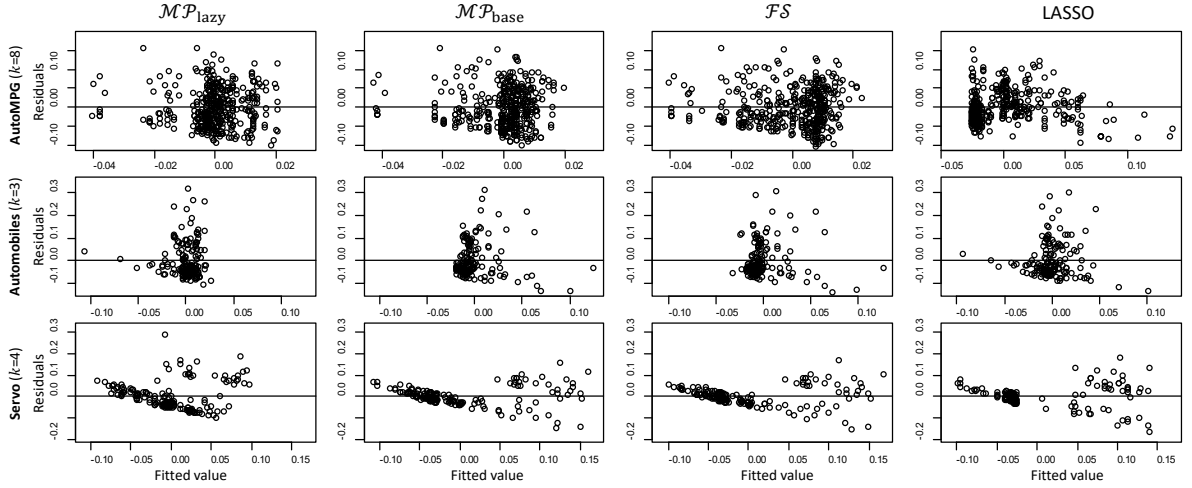


Figure 5: Residual plots for three representative cases

Results for Experiment 2: Proposed model vs. iterative model In Table 7, we compare our model with the iterative model ($\mathcal{MP}_{\text{iter}}$) for cases where the solution for $\mathcal{MP}_{\text{base}}$ violates some of the tests and diagnostics. We only consider these 28 cases because both algorithms are not needed if $\mathcal{MP}_{\text{base}}$ can provide a solution that satisfies all tests and diagnostics. In the Status column, ‘alt sol’ indicates that, within the time limit, a model could not find a solution with statistically significant coefficients for

Table 6: p -values from heteroscedasticity tests

Case (dataset, k)	r_h			
	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{base}}$	\mathcal{FS}	LASSO
AutoMPG ($k = 8$)	0.0142	2.6402×10^{-5}	2.7821×10^{-8}	2.8149×10^{-12}
Automobile ($k = 3$)	0.0111	3.3471×10^{-6}	1.1835×10^{-9}	1.6423×10^{-6}
Servo ($k = 4$)	0.0106	1.4686×10^{-11}	1.5836×10^{-16}	7.1300×10^{-24}

all of the selected variables, and an alternative solution is obtained at termination. The terms ‘opt’ and ‘best feas’ represent cases where an optimal solution (satisfying all constraints) and a feasible solution are found, respectively. The term ‘infeas’ for $\mathcal{MP}_{\text{lazy}}$ means that the problem is infeasible and our model returns an alternative solution. These cases are organized according to the following hierarchy: ‘opt’ > ‘best feas’ > ‘alt sol’ = ‘infeas’. In columns Status, E , and π , the results indicating that our model is better are in boldface. The results in Table 7 are summarized in Table 8.

Table 7: Comparative results for the iterative model

dataset	k	Status		REP		E		π		Execution time	
		$\mathcal{MP}_{\text{iter}}$	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{iter}}$	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{iter}}$	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{iter}}$	$\mathcal{MP}_{\text{lazy}}$	$\mathcal{MP}_{\text{iter}}$	$\mathcal{MP}_{\text{lazy}}$
Servo	10	alt sol	best feas	1	0.9963	0.1745	0	1	0	3600	3601
Bodyfat	3	opt	opt	0.9998	0.9998	0	0	0	0	83	5
Bodyfat	4	alt sol	opt	0.9997	0.7557	0.2541	0	1	0	3600	86
Bodyfat	5	alt sol	opt	0.9999	0.7539	0.162	0	4	0	3600	1978
Bodyfat	6	alt sol	best feas	0.9999	0.7568	0.3183	0	5	0	3600	3601
Bodyfat	7	alt sol	best feas	0.9999	0.7414	0.343	0	6	0	3600	3601
Bodyfat	8	alt sol	alt sol	1	0.7524	0.2904	0.1619	7	1	3600	3601
Bodyfat	9	alt sol	alt sol	1	0.7587	0.2898	0.1403	8	4	3600	3601
Bodyfat	10	alt sol	alt sol	1	0.76	0.3554	0.3026	9	3	3600	3601
Barro	6	alt sol	alt sol	0.4900	0.5191	0.5495	0.0931	3	1	3600	3600
Barro	7	alt sol	alt sol	0.9664	0.5372	0.3176	0.1095	3	2	3600	3601
Barro	8	alt sol	alt sol	0.9829	0.5171	0.4093	0.1817	3	3	3600	3600
Barro	9	alt sol	alt sol	0.9905	0.9469	0.4361	0.2407	4	4	3600	3600
Barro	10	alt sol	alt sol	0.9892	0.9721	0.391	0.2507	5	5	3600	3601
Winequality	8	alt sol	opt	0.9959	0.9924	0.4756	0	1	0	3600	88
Winequality	9	alt sol	opt	0.9988	0.9974	0.5346	0	2	0	3600	48
Winequality	10	alt sol	alt sol	0.9988	0.9863	0.4835	0.0725	3	2	3600	3600
Carseats	8	opt	opt	0.9998	0.9998	0	0	0	0	397	9
Carseats	9	alt sol	opt	0.8117	0.9923	0.5056	0	3	0	3600	289
Carseats	10	alt sol	alt sol	1	0.9502	0.2099	0.1743	2	1	3600	3601
Framing	7	alt sol	alt sol	0.9886	0.9971	0.2128	0.1768	2	1	3600	3601
Framing	8	alt sol	alt sol	0.9643	0.8653	0.2711	0.1649	4	3	3600	3602
Framing	9	alt sol	alt sol	0.9965	0.8651	0.2213	0.1614	4	4	3600	3601
Framing	10	alt sol	alt sol	0.9969	0.9654	0.3761	0.1957	5	5	3600	3601
Griliches	9	alt sol	opt	1	0.9963	0.0509	0	1	0	3600	417
Griliches	10	alt sol	opt	0.9996	0.9934	0.0604	0	2	0	3600	843
Hprice3	9	opt	opt	0.9952	0.9952	0	0	0	0	625	47
Hprice3	10	alt sol	alt sol	0.9914	0.9936	0.553	0.0943	3	1	3600	3600
Average	-	-	-	0.9698	0.8699	0.2945	0.0900	3.25	1.43	3253.7	2450.9

Table 8: Summary table for the results in Table 7

Measures	$\mathcal{MP}_{\text{iter}}$ wins	$\mathcal{MP}_{\text{lazy}}$ wins	Tie
Status	0	10	18
E	0	25	3
π	0	20	8
Execution time	0	10	18

According to Tables 7 and 8, $\mathcal{MP}_{\text{lazy}}$ outperforms $\mathcal{MP}_{\text{iter}}$ in most cases, while maintaining an explanatory power around 87% of $\mathcal{MP}_{\text{base}}$. In cases where $\mathcal{MP}_{\text{lazy}}$ finds an optimal solution, $\mathcal{MP}_{\text{lazy}}$ is appreciably faster than $\mathcal{MP}_{\text{iter}}$. Moreover, by comparing performance measures E and π for the alternative solutions, we can see that the alternative solutions from our model outperform those from the iterative model in terms of the statistical significance of the coefficients, while differences in the explanatory power can be ignored in most cases. Furthermore, for Barro ($k = 6$), Carseats ($k = 9$), Framing ($k = 7$), and Hprice3 ($k = 10$), our model finds a better solution in terms of both explanatory power and statistical significance. Several results in the Bodyfat and Barro datasets indicate that the explanatory power of our model is less than 80% of $\mathcal{MP}_{\text{base}}$. However, $\mathcal{MP}_{\text{lazy}}$ clearly outperforms $\mathcal{MP}_{\text{iter}}$ in terms of the statistical significance of the coefficients in these cases.

Results for Experiment 3: Proposed model vs. model without the relaxed t -test constraints

To confirm the effectiveness of the proposed relaxed t -test constraint (8g), we compare our proposed model with and without (8g). We run the models on the cases (dataset and k pairs) where the optimal solution is found within 600 seconds. For each case, we run the algorithms 30 times to obtain the average and median number of explored nodes and computation time. Table 9 provides summary statistics by aggregating the results by k , where the complete results are reported in the online supplement.

Table 9: Summary table for the results of experiment 3

		#node		Time	
k	#case	Ratio	%improved	Ratio	%improved
3	12	0.998	72.7	1.051	18.2
4	12	0.984	63.6	1.015	27.3
5	5	0.862	80.0	0.940	60.0
6	5	0.821	100	0.892	100
7	2	0.918	100	0.972	50
8	2	0.831	100	0.893	50
ALL	39	0.936	77.8	0.989	41.7
$k \geq 5$	14	0.846	92.3	0.921	71.4

For each case (dataset and k pair), we define Ratio to be the average ratio between the model with and without the cut. If Ratio is less than 1, it implies that the model with the cut is better for the corresponding performance measure (either #node and Time). In Table 9, for each k , Ratio is the average Ratio out of all corresponding cases and %improved is the percentage of cases with Ratio < 1 . The last two rows include the averages across all cases and cases with $k \geq 5$.

In Table 9, we can verify the advantage of including the relaxed t -test constraint. For every k , the Ratio for #node is less than 1. This shows that our model searches less number of solutions before obtaining the optimal solution than the model without the constraint. This is because the constraint reduces the solution space by cutting some solutions with insignificant coefficients. As a result, the computation time to obtain the optimal solution decreases as shown in the Time column. In particular, the computation time of the model with the cut is significantly faster when k is larger (e.g., $k = 5, \dots, 8$): by including the constraint, #node and computation time are significantly decreased by 15.4% and 7.9%, respectively. This trend implies that for a larger k , more solutions are excluded by the constraint. It corresponds to the fact that it is more likely to have insignificant coefficients when we have more explanatory variables.

Results for Experiment 4: Alternative solution procedure vs. simple penalty function

In Table 10, the p -values for the coefficients from the linear models derived from $\mathcal{MP}_{\text{lazy}}$ and $\mathcal{MP}_{\text{penalty}}$ for Bodyfat ($k = 7$) and Carseat ($k = 10$) are presented, where the two cases are selected because the two approaches provide clearly distinguished solutions. Recall that the only difference between $\mathcal{MP}_{\text{lazy}}$ and $\mathcal{MP}_{\text{penalty}}$ is that the former uses the alternative solution procedure whereas the latter uses the simple penalty function f_q . The p -values greater than $1 - \alpha_E = 1 - 0.95 = 0.05$ are bolded.

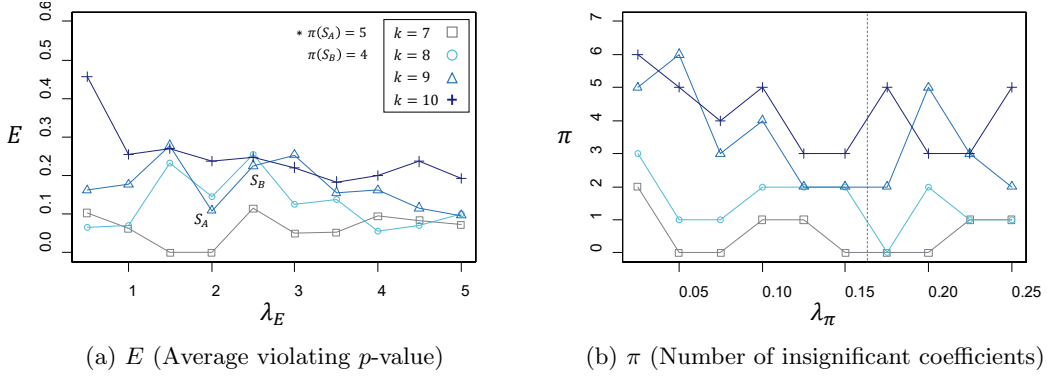
Note that the average of the violating p -values (E) for $\mathcal{MP}_{\text{lazy}}$ for Bodyfat ($k = 7$) is 0.0649, while it is 0.0899 for $\mathcal{MP}_{\text{penalty}}$. Also, the R^2_{adj} of $\mathcal{MP}_{\text{lazy}}$ is greater than that of $\mathcal{MP}_{\text{penalty}}$. In Carseat ($k = 10$), the average of the violating p -values for $\mathcal{MP}_{\text{lazy}}$ is smaller than that of $\mathcal{MP}_{\text{penalty}}$ ($0.1736 < 0.1800$) and furthermore the number of insignificant coefficients for $\mathcal{MP}_{\text{lazy}}$ is smaller than that of $\mathcal{MP}_{\text{penalty}}$ ($1 < 2$). The R^2_{adj} of $\mathcal{MP}_{\text{lazy}}$ is worse, however, the difference in R^2_{adj} is negligible. Thus, our model,

Table 10: p -values from the t -tests for the coefficients

Dataset (k)	Model	R_{adj}^2	E	p -values									
Bodyfat (7)	$\mathcal{MP}_{\text{lazy}}$	0.7288	0.0649	0.0000	0.0649	0.0112	0.0001	0.0076	0.0045	0.0091			
	$\mathcal{MP}_{\text{penalty}}$	0.7274	0.0899	0.0899	0.0000	0.0035	0.0402	0.0302	0.0050	0.0179			
Carseat (10)	$\mathcal{MP}_{\text{lazy}}$	0.8296	0.1736	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000	0.1736	0.0000	0.0000	0.0000
	$\mathcal{MP}_{\text{penalty}}$	0.8682	0.1800	0.0000	0.0000	0.0000	0.0000	0.0004	0.1800	0.0000	0.0000	0.0000	0.1800

which includes the alternative solution procedure, can provide a better solution than the model employing a simple penalty function.

Results for Experiment 5: Sensitivity analysis The plots in Figure 6 represent the trends in E and π according to changes in λ_E and λ_π for the Bodyfat dataset, respectively. In each plot, it can be observed that each violation measure decreases with an increase in the corresponding parameter. This relationship does not appear to be monotonous, because the optimization algorithm embedded in the solver searches different spaces and terminates due to the time limit every time it is called. For example, let us compare the solutions S_A and S_B in Figure 6a. (The other validation measures are not significantly different.). According to each value of E and π , it is obvious that S_A is a better solution than S_B under the parameter set with $\lambda_E = 2$ and $\lambda_E = 2.5$. However, because the optimization solver did not find S_A during the search with $\lambda_E = 2.5$, S_B is returned as the best alternative solution. In Figure 6b, it is noteworthy that the decrease in π does not appear when λ_π is greater than 0.15. This indicates that increasing the parameter does not necessarily improve the validation measure after a certain point.


 Figure 6: Trends in the measures according to changes in the parameters (Bodyfat with $k = 7, 8, 9, 10$)

6 Conclusion

Although minimizing fitting errors is the most important objective when building a regression model, it is also important to check if all the regression assumptions and diagnostics are met or satisfied. However, most of the current approaches do not (or cannot) incorporate this model validation step into their algorithms. In this work, we propose a fully automated model building procedure for multiple linear regression subset selection that integrates model building and validation based on mathematical programming. The procedure is further improved in terms of computation efficiency by including the relaxed t -test constraint which can be efficiently derived. The proposed procedure is also capable of providing an alternative solution when there is no feasible solution or a feasible solution is not found within the set time limit.

The computational experiments with real-world datasets show that our model can provide quality solutions satisfying all of the considered diagnostics while maintaining an R_{adj}^2 value close to the base solution (optimal value when all the diagnostics are ignored). Furthermore, the results demonstrate the viability of our model in real regression analysis by setting a practical time limit. We also find that the alternative solutions are superior to those of the benchmarks. The R_{adj}^2 value of the alternative solutions

are comparable with those of the benchmarks, while being significantly better in terms of diagnostics. We also show that our model based on lazy constraints outperforms the existing iterative approach. The proposed lazy constraint-based procedure is faster than the iterative approach while providing more quality alternative solutions within the time limit. Lastly, we demonstrate the benefit of the relaxed t -test constraint showing that including the constraint can significantly decrease the number of explored solutions and the computation time.

In conclusion, we strongly believe that our model is useful in practice because we integrate model building (subset selection) and validation steps, whereas traditional approaches require human decisions and significant trial-and-error to alternate the two steps. Furthermore, the proposed lazy constraint approach based on mathematical programming that addresses the statistical tests and diagnostics for linear regression is substantially more efficient than the existing iterative model. We also remark that other regression diagnostics that are not considered in this paper can be easily integrated into our lazy constraint framework. One of the most challenging research directions for the improvement of the current procedure includes formulating a linear or a convex constraint to replace the current approximation constraint for t -tests or improve the quality of the approximated constraint.

Acknowledgements

This research was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (Ministry of Science, ICT & Future Planning) (No. NRF-2015R1C1A1A02036682) and was also supported in part by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (The Competency Development Program for Industry Specialist) under Grant P0008691.

References

- [1] D. Bertsimas and A. King. OR forum - an algorithmic approach to linear regression. *Operations Research*, 61(1):2–16, 2016.
- [2] D. Bertsimas and M. L. Li. Scalable holistic linear regression. *Operations Research Letters*, 2020.
- [3] D. Bertsimas, A. King, and R. Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.
- [4] M. Blackburn and D. Neumark. Unobserved ability, efficiency wages, and interindustry wage differentials. *The Quarterly Journal of Economics*, 107(4):1421–1436, 1992.
- [5] T. Brader, N. A. Valentino, and E. Suhay. What triggers public opposition to immigration? anxiety, group cues, and immigration threat. *American Journal of Political Science*, 52(4):959–978, 2008.
- [6] T. S. Breusch and A. R. Pagan. A simple test for heteroscedasticity and random coefficient variation. *Econometrica: Journal of the Econometric Society*, 47(5):1287–1294, 1979.
- [7] E. Carrizosaa, A. V. Olivares-Nadal, and P. Ramirez-Cobob. Novel constraints for enhancing interpretability in linear regression. Technical report, 2017.
- [8] G. Castellano and A. M. Fanelli. Variable selection using neural-network models. *Neurocomputing*, 31(1):1–13, 2000.
- [9] C. Cornwell and W. N. Trumbull. Estimating the economic model of crime with panel data. *The Review of Economics and Statistics*, 76(2):360–366, 1994.
- [10] R. Genuer, J. M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.
- [11] E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

- [12] A. Gómez and O. Prokopyev. A mixed-integer fractional optimization approach to best subset selection. *Optimization-online*, 2018.
- [13] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016. URL <http://www.gurobi.com>.
- [14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.
- [15] F. Hafiz, A. Swain, N. Patel, and C. Naik. A two-dimensional (2-d) learning framework for particle swarm based feature selection. *Pattern Recognition*, 76:416–433, 2018.
- [16] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*, volume 6. Springer, 2013.
- [17] R. W. Johnson. Fitting percentage of body fat to simple body measurements. *Journal of Statistics Education*, 4(1):265–266, 1996.
- [18] A. G. Karegowda, M.A. Jayaram, and A.S. Manjunath. Feature subset selection problem using wrapper approach in supervised learning. *International Journal of Computer Applications*, 1(7): 13–17, 2010.
- [19] Y. Kim and S. B. Kim. Collinear groupwise feature selection via discrete fusion group regression. *Pattern Recognition*, 83:1–13, 2018.
- [20] R. Koenker and J. A. F. Machado. Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association*, 94(448):1296–1310, 1999.
- [21] H. Konno and Y. Takaya. Multi-step methods for choosing the best set of variables in regression analysis. *Computational Optimization and Applications*, 46(3):417–426, 2010.
- [22] H. Konno and R. Yamamoto. Choosing the best set of variables in regression analysis using integer programming. *Journal of Global Optimization*, 44(2):273–282, 2009.
- [23] M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- [24] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.
- [25] R. Miyashiro and Y. Takano. Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, 247(3):721–731, 2015.
- [26] R. Miyashiro and Y. Takano. Subset selection by mallows’ C_p : A mixed integer programming approach. *Expert Systems with Applications*, 42(1):325–331, 2015.
- [27] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*, volume 4. Irwin Chicago, 1996.
- [28] Y. W. Park and D. Klabjan. Subset selection for multiple linear regression via optimization. *Journal of Global Optimization*, 77:543–574, 2020.
- [29] A. Rakotomamonjy. Variable selection using SVM-based criteria. *Journal of machine learning research*, 3(Mar):1357–1370, 2003.
- [30] A. C. Rencher and G. B. Schaalje. *Linear models in statistics*. John Wiley & Sons, 2008.
- [31] T. Sato, Y. Takano, R. Miyashiro, and A. Yoshise. Feature subset selection for logistic regression via mixed integer optimization. *Computational Optimization and Applications*, 64(3):865–880, 2016.
- [32] S. Seabold and J. Perktold. Statsmodels: Econometric and statistical modeling with Python. In *9th Python in Science Conference*, 2010.
- [33] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.

- [34] R. Tamura, K. Kobayashi, Y. Takano, R. Miyashiro, K. Nakata, and T. Matsui. Best subset selection for eliminating multicollinearity. *Journal of the Operations Research Society of Japan*, 60(3):321–336, 2017.
- [35] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [36] H. G. Wilson. Least squares versus minimum absolute deviations estimation in linear models. *Decision Sciences*, 9(2):322–335, 1978.
- [37] J. M. Wooldridge. *Introductory econometrics : a modern approach*, volume 6. South-Western Cengage Learning, 2006.
- [38] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5(Oct):1205–1224, 2004.
- [39] H. Zhang and G. Sun. Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recognition*, 35(7):1481–1490, 2002.
- [40] K. Zheng and X. Wang. Feature selection method with joint maximal information entropy between features and class. *Pattern Recognition*, 77:20–29, 2018.